

MANUAL CHANGES

MANUAL IDENTIFICATION

Model Number: 10269A

Date Printed: OCTOBER 1983

Part Number: 10269-90905

This supplement contains important information for correcting manual errors and for adapting the manual to instruments containing improvements made after the printing of the manual.

To use this supplement:

Make all ERRATA corrections.

Make all appropriate serial number related changes indicated in the tables below.

Serial Prefix or Number	Make Manual Changes

Serial Prefix or Number	Make Manual Changes

▲ NEW ITEM

▲ ERRATA

Add: Appendix I (Pages I-1 through I-10) to the manual.

Add: Appendix J (Pages J-1 through J-17) to the manual.

NOTE

Manual change supplements are revised as often as necessary to keep manuals as current and accurate as possible. Hewlett-Packard recommends that you periodically request the latest edition of this supplement. Free copies are available from all HP offices. When requesting copies quote the manual identification information from your supplement, or the model number and print date from the title page of the manual.

6809/6809E

The Model 10269A Option 071 is the 10269A General Purpose Probe Interface and the Model 64671A Preprocessor Interface Module. When used together, a complete interface is provided between any 6809 or 6809E target system and the Model 1630A/D Logic Analyzer. The interface module connects the signals from the 6809/6809E target CPU to the logic analyzer inputs and generates all status and clock signals required by the software for inverse assembly of the 6809/6809E instruction set.

Processor Compatibility:	Motorola 6809/6809E and all microprocessors made by other manufacturers which comply with the Motorola 6809/6809E specifications.
Maximum Clock Speed:	2 Mhz clock output from microprocessor.
Signal Line Loading:	One ALS TTL load for all monitored lines plus approximately 35 pf capacitance.
Power Consumption:	1.0 A at +5Vdc max, supplied by Model 1630A/D Logic Analyzer.
Environmental	
Temperature:	Operating, 0 to 55 degrees C (+32 to +131 degrees F); Nonoperating, -40 to +75 degrees C (-40 to +167 degrees F).
Altitude:	Operating, 4600 m (15 000 ft); Nonoperating, 15 300m (50 000 ft).
Humidity	To 90% noncondensing. Avoid sudden, extreme temperature changes which could cause condensation within the instrument.

System Configuration

The minimum requirements for performance analysis of a 6809/6809E target system consist of the following hardware:

- a. Model 1630A/D Logic Analyzer (with HP 82161A Digital Cassete Drive for inverse assembly).
- b. 10269A General Purpose Probe Interface with Option 071 which includes the following items:
 - (1) 64671A Interface Module for 6809/6809E micro-processors, including a 6809/6809E interface circuit card and cable assembly.
 - (2) 64671A inverse assembler and Format Specification setups on Minicassette HP Part No 64671-11000.
- c. BNC cable assembly to connect +5V accessory power from logic analyzer to general purpose probe interface.

The Model 1630D enables storage of up to 8 additional channels of information.

Hardware Installation

Install the interface module on the underside of the general purpose probe interface pod. Insert the free end of the interface card into the slots of the pod, then connect the pod internal cables to the interface circuit card, and fasten the cable end of the interface module to the pod using two screws.

Connect a BNC cable from the ACCESSORY POWER output on the rear panel of the logic analyzer to the ACCESSORY POWER input on the general purpose probe interface.

NOTE

Some interface modules can draw up to 1 ampere from the 5-volt supply. If using coaxial cable, HP recommends that you use RG-58, 4-foot or less to minimize voltage drop.

Connect the interface cable to the target system by lifting the 6809/6809E CPU from its socket, plugging the interface cable into the CPU socket, and plugging the CPU into the interface cable. Be sure pin 1 of the CPU is installed in pin 1 of the cable socket. Pin 1 of the cable socket is identified on the CPU socket board and shown in figure A-1.

The CPU socket on the interface cable is a Zero Insertion Force (ZIF) socket. Before installing the CPU, open the socket by rotating the small screw located at the pin 1 end of the socket. To secure the CPU in the socket, rotate the screw in the opposite direction.

To obtain a lower profile assembly, the probe cable can be used without the ZIF socket. Figure A-1 is an exploded view of the socket assembly. Separate the ZIF socket assembly by inserting a flat blade screw driver under the pc board located directly below the ZIF socket and carefully prying up. Avoid bending the pins of the ZIF socket or probe cable assembly. Plug the probe assembly into the target system, and install the CPU in the cable socket. The plastic pin guard can be removed if it interferes with components of the target system, or if an even lower profile is required. Removal of the pin guard is not recommended, however, because this increases the danger of damage to the probe pins.

General Purpose
Probe Interface

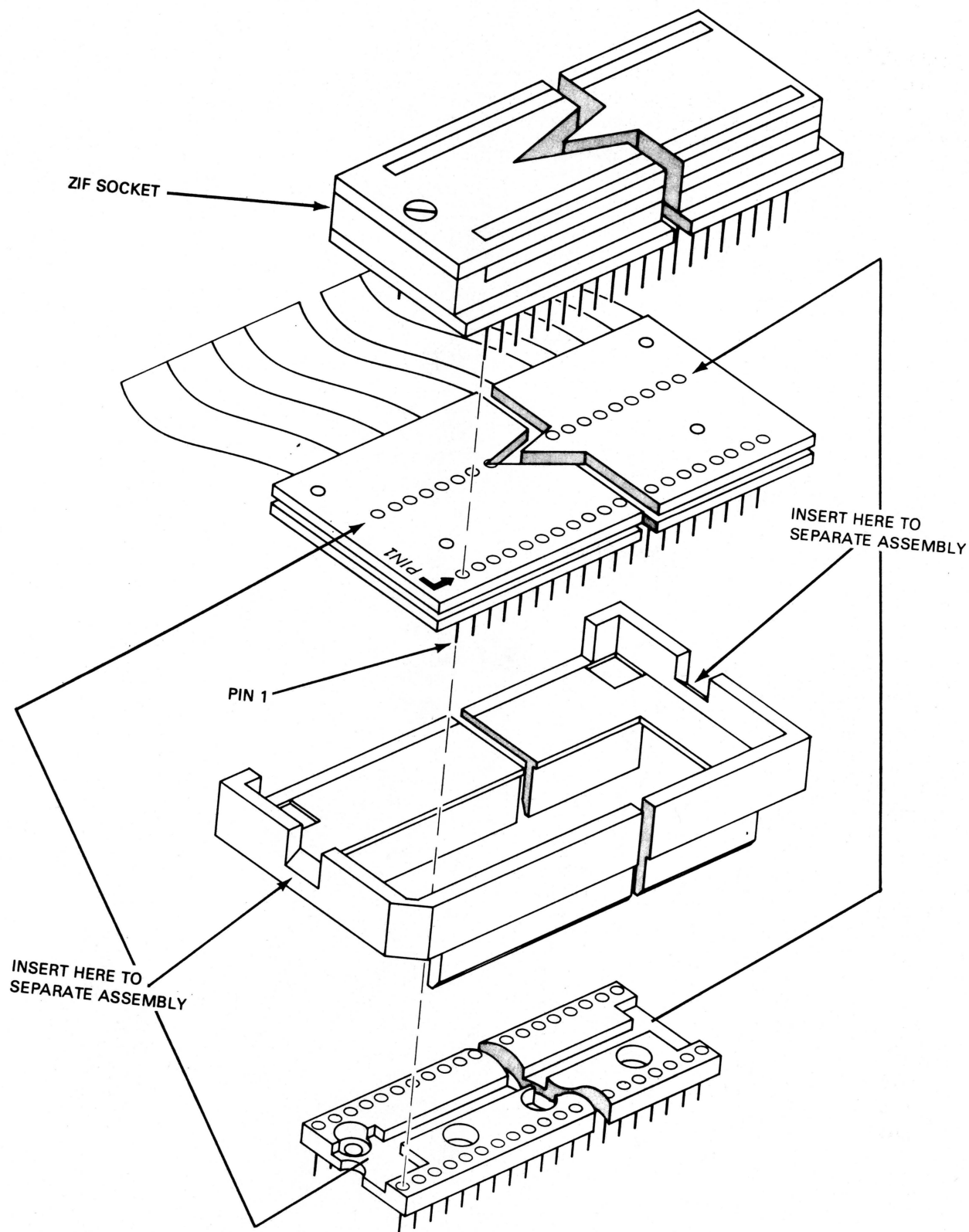


Figure I-1. ZIF Socket Assembly, Exploded View

General Purpose Probe Interface

Connect the 1630A/D Logic Analyzer pods to the general purpose interface (Model 10269A) as shown in the following list:

Model 10269A Connector	Model 1630A/D Pod
A	1
B	4
C	2
D	3
E	n/c

Interface Description

The primary function of the interface module is to connect the target CPU to the logic analyzer through the general purpose probe interface and to perform any required functions that are unique to the 64671A 6809/6809E Interface Module.

The interface tracks bus cycles of the microprocessor, decodes instructions, and generates status information about each bus cycle. Figure I-2 is a block diagram of the 64671A Interface Module. Table I-1 is a list of the CPU signals and corresponding lines to the logic analyzer.

Signals from the target system are sampled at the end of each cycle at the Bus Latch. The Control Logic section receives as inputs the E and Q clock signals and the BA, BS, and R/W status lines from the CPU and produces several signals. First, two signals are generated to clock the state analyzer. The J clock is the signal normally used; it is active for each processor cycle that is useful for analysis. The determination of these cycles is made by the Status Generator. The K clock is active for every cycle of the E and Q clocks. The negative edge of both J clock and K clock is used to clock the logic analyzer; for a given bus cycle this will occur at the negative edge of Q on the following bus cycle.

Status Generation

Another function of the interface module is to generate status information about processor activity which is not available from the processor itself. This additional status is generated by a state machine on the interface that tracks all bus cycles and decodes instructions. The type of status generated identifies opcodes, other code bytes, data transfers, unused bus cycles, illegal instructions, and other types of cycles. Using status generated by the interface is described in the last section of this appendix.

Target System Load

The interface module was designed to place a minimum electrical load on the target system. There are a few things pertaining to this loading that are important. First, all lines that are monitored by the interface will load the target system with one ALS TTL load and approximately 35 pf of capacitance.

Those lines include the address and data bus, BA, BS, R/W status lines, and E and Q clock lines. The two pins that connect to the 6809 on-chip oscillator (CPU pins 38 and 39) have only a one or two pf capacitive load and should not interfere with the oscillator. All other signal carrying lines are connected to the probe cable, but not to any circuitry on the interface module, and have a load of approximately 25 pf.

To protect the interface from damage caused by static discharge, clamping diodes have been provided on the 64671A interface board. These diodes clamp the input voltages from the probe cable to the +5 volt power supply and to ground at the preprocessor. Because of these diodes, 6809/6809E signals will be clamped to ground if the logic analyzer and interface are not powered up, and the target system will be prevented from operating.

If the logic analyzer cannot always be powered-up when the target system is operating, the diode arrays (U1A, U2A, U2B, and U3A) may be removed from their sockets to allow CPU operation. However, removal of the diode arrays will increase interface module susceptibility to damage from static discharge.

General Purpose
Probe Interface

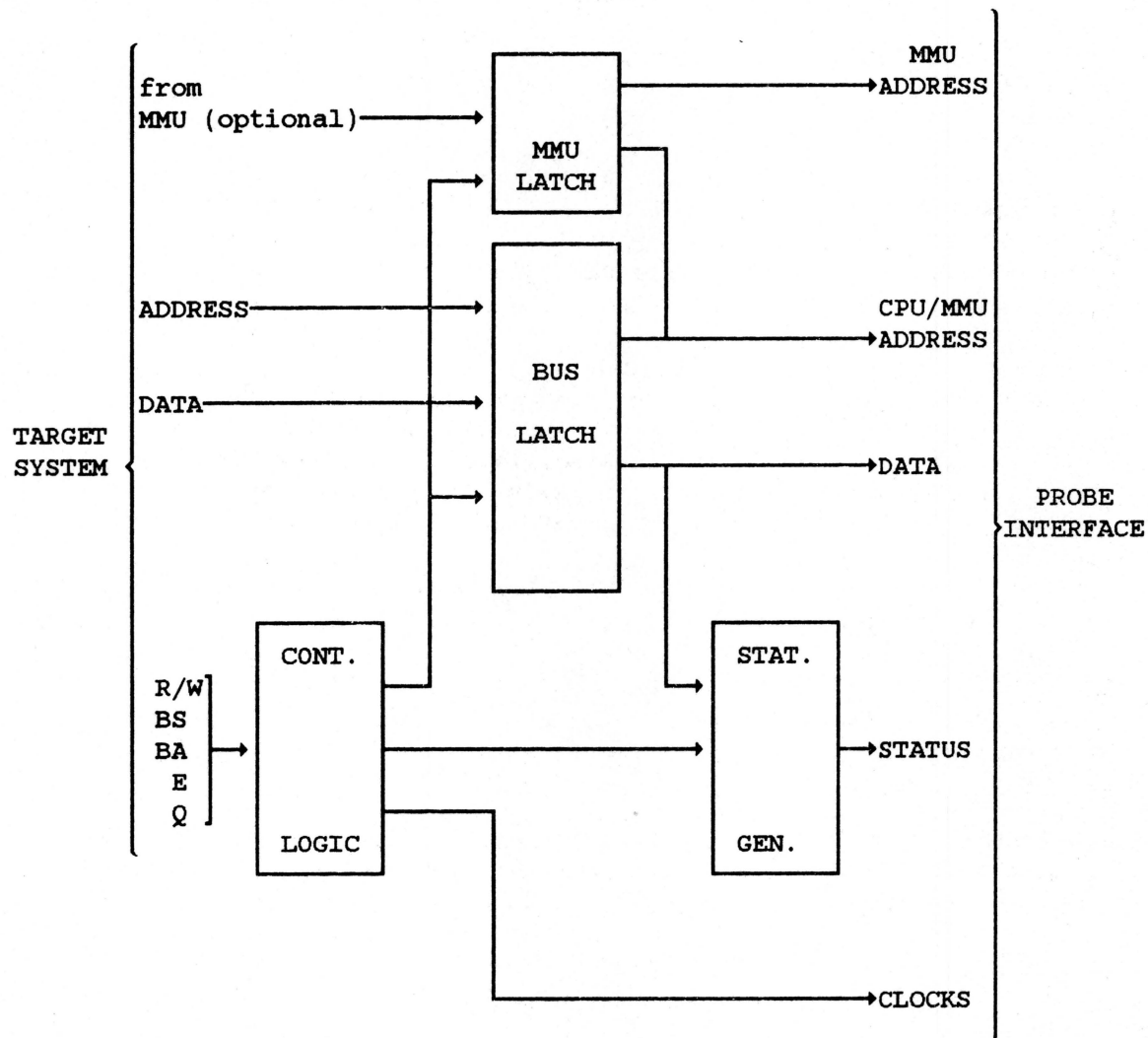


Figure I-2. 6809/6809E Block Diagram

Table I-1. 6809/6809E Signal List

CPU SIGNAL	CPU PIN	LABEL	POD	BIT
NOTE 1		(CLOCK)	4	J CLK
Q	35	(CLOCK)	3	K CLK
A0	8	ADDR	2	0
A1	9	ADDR	2	1
A2	10	ADDR	2	2
A3	11	ADDR	2	3
A4	12	ADDR	2	4
A5	13	ADDR	2	5
A6	14	ADDR	2	6
A7	15	ADDR	2	7
A8	16	ADDR	2	8
A9	17	ADDR	3	0
A10	18	ADDR	3	1
A11	19	ADDR	3	2
A12	20	ADDR	3	3
A13	21	ADDR	3	4
A14	22	ADDR	3	5
A15	23	ADDR	3	6
D0	31	DATA	1	0
D1	30	DATA	1	1
D2	29	DATA	1	2
D3	28	DATA	1	3
D4	27	DATA	1	4
D5	26	DATA	1	5
D6	25	DATA	1	6
D7	24	DATA	1	7
R/LW	32	STAT	4	0
BS	5	STAT	4	1
BA	6	STAT	4	2
NOTE 2		STAT	4	3
NOTE 2		STAT	4	4
NOTE 2		STAT	4	5
NOTE 2		STAT	4	6
NOTE 2		STAT	4	7

NOTE 1: This clock line has timing similar to the 6809 Q line but is active only for cycles that are needed for state analysis.

NOTE 2: These five status lines are generated by circuitry in interface module.

Status Field Encoding

There are eight bits in the complete status field that may be divided into two groups. The lower three bits are status lines that are available from the 6809/6809E. The upper five bits are generated by circuitry on the interface module. The eight bits have the following definitions:

Bit 7 = 0 for cycles that are needed for analysis of
6809/6809E instruction execution
= 1 for cycles that are non-valid, fetch ahead, sync
acknowledge, halt or non-processor cycles

Bit 6 is always low

Bit 5 thru 3 are generated by the interface module and are
encoded together for the following actions:

000 = first byte of an instruction (opcode)
001 = second byte or following byte of an instruction
010 = indirect addressing mode pointer fetch
011 = data read or write as the result of instruction
execution
100 = illegal first byte of an instruction
101 = illegal second or following instruction byte
110 = instruction interrupted by hardware interrupt
(this status is only given on the second cycle of
an instruction)
111 = no status generated due to the fetch of an illegal
instruction (status generation will resume with a
processor write or reset/interrupt vector read cycle)

Bit 2 is the 6809/6809E BA line

Bit 1 is the 6809/6809E BS line

Bits 2 and 1 are encoded for the following states:

00 = normal (processor executing instructions)
01 = vector fetch, reset, or interrupt
10 = sync acknowledge
11 = halt or bus grant

Bit 0 is the 6809/6809E R/W line

0 = write cycle
1 = read cycle

Operation

The 1630 A/D Logic Analyzer will be configured for state analysis when the inverse assembler is loaded from the digital cassette drive.

6809/6809E Format Specification

When using the 6809/6809E interface module, the format specification will automatically be set up by the mini-cassette software as in figure I-3.

```

State Format Specification_____

[Assignment]                                Clock
                                           JKL
                                           -----[↓..]-----

Multiplexing      Pod 4      Pod 3      Pod 2      Pod 1      Pod 0
  [Off]          [ TTL ] [ TTL ] [ TTL ] [ TTL ] [ TTL ]
                8.....0 8.....0 8.....0 7.....0 7.....0

Activity>
Label Pol
ADDR  [+ ] [ ..... ..***** ***** ]
DATA  [+ ] [ ..... ..***** ***** ]
STAT  [+ ] [ ..... ..***** ***** ]

```

Figure I-3. 6809/6809E Format Specification

Inputs to the logic analyzer are grouped under functional labels for the convenience of the user. The ADDR, DATA, and STAT labels are used by the inverse assembler to generate assembly language mnemonics for display in the trace list. Pertinent control and clock threshold levels are also part of the format specification. The J clock is normally used to clock the logic analyzer. This signal will be active for all 6809 cycles that are useful for state analysis.

No unused or fetch ahead cycles made by the CPU will be sent to the analyzer. Sync acknowledgement and halt/bus grant cycles will not be sent to the analyzer. The K clock is the 6809 Q line after being buffered. This clock should be used when it is desired to see bus cycles that are not included in the J clock.

The 6809/6809E Inverse Assembler

The 6809/6809E inverse assembler has been constructed so mnemonic output will resemble actual assembly source code. Numerical values such as addresses and immediate values are normally displayed in hexadecimal format. The one exception to this is the immediate value for the ANDCC and ORCC instructions that are displayed in binary form, making easier the determination of which bits are being modified in the condition code register.

A pair of asterisks (**) displayed in the operand field of an instruction indicates that a byte of an operand was not stored in analyzer memory. Two pairs of asterisks (****) indicate that two bytes of an expected operand are missing.

Note that missing operands (or parts of operands) may be the result of storage qualification.

Examples:

LDA	***	(missing byte operand)
JMP	23**	(missing low byte of address)
JMP	****	(missing both bytes of address)

Error Messages

The following messages will appear in the case of program errors.

- Illegal instruction - unidentified opcode encountered on the first byte, or on the second byte if the first byte is 10H or 11H
- Illegal postbyte - the register mask for a transfer or exchange instruction, or the indexed addressing mode is illegal
- Undefined bus cycle - the combination of bus available (BA), bus status (BS), and read/write (R/W) is not recognized for the 6809/6809E

NOTE

No modification of the ADDR, DATA, or STAT labels in the format specification should be made if inverse assembly is desired. Any changes may cause incorrect results.

Using Status Generated By The Interface

When status from the processor indicates normal running, i.e., BA=0, BS=0, the only other status indication given by the processor is whether the transfer is to or from memory. What this status does not identify is whether the transfer is an opcode fetch, a subsequent code byte, or a data transfer due to instruction execution. The missing information is generated, however, by circuitry on the interface module and provided to the logic analyzer along with the address and data for each bus cycle. This additional status information is particularly useful in the analysis of program activity because it is generated in real time and allows the state analyzer to trigger on or store qualify on various types of bus cycles that are not identified by the processor.

Status Labels

There are five different status labels that the interface module uses to identify most bus cycles made by the 6809/6809E processor. The five types of cycles have the following definitions:

- Opcode - the first byte of any instruction
- Program - the second or following byte of any instruction
- Pointer - the two byte pointer fetched during an indirect addressing mode instruction
- Data - the memory read or write due to program execution
- Unused - a cycle which has no value for the logic analyzer

By providing this additional status information, defining trace specifications is easier and more specific. For example, the analyzer can be triggered on fetching an opcode from an address that is outside the known legal range for instructions. With a "center on" or "end on" trace specified, the cause of the microprocessor's illegal program transfer can be determined. By storing only opcodes, a much larger picture of program activity is captured on a single trace.

When a bus cycle that has no value for state analysis is encountered during normal program execution, the status label "unused" is issued to the logic analyzer. This information is normally not stored in the analyzer memory. These unused cycles are withheld from the logic analyzer by clocking the analyzer with the J clock. The falling edge of the K clock must be used when it is necessary to see the "unused" cycles. The following paragraphs identify some of the types of "unused" cycles that may occur.

The microprocessor performs either a memory read or write for each cycle of the E clock; some of the memory reads are redundant. As the processor is fetching an instruction, it may fetch one or more bytes of the next instruction. The bytes are fetched before the instruction has been completely decoded, therefore, some of the bytes are unnecessary and are discarded. The discarded bytes will be fetched again if that next instruction is executed. The interface module identifies the discarded

General Purpose Probe Interface

bytes as "unused" cycles. An "unused" cycle prevents a read to a legitimate address in the analyzer memory.

A second type of unused bus cycle occurs when the microprocessor is performing an internal operation which takes one or more clock cycles. During this time the processor will read from memory location OFFFFH. The interface module identifies these cycles as "unused". This identification is not based solely on the address OFFFFH and a valid data read from address OFFFFH will not be labeled as "unused".

There is another type of cycle which is classified as unused by the interface. If an instruction begins with either 10H or 11H, there must be a second byte to define which instruction is to be executed. If the second byte is also 10H or 11H, the processor ignores the data value and continues reading from memory until a value which is not 10H or 11H is found for the second instruction byte. If a string of 10H or 11H data values occurs, such as in a dynamic memory refresh subroutine, the second and following values of 10H or 11H are labeled as "unused" by the status generator on the interface.

Illegal and Interrupted Instructions

The status generator on the interface module will detect and identify illegal instructions that occur. Instructions that can be determined as illegal on the first byte will be labeled as illegal opcodes. Instructions that begin with 10H or 11H but have an illegal second byte will have the first byte labeled as an opcode, but the second byte will be labeled as an illegal program byte. If the postbyte that specifies the addressing mode in an indexed addressing mode instruction is illegal, then the postbyte will be given a status value of illegal program. Illegal postbytes in transfer and exchange instructions are not detected.

If the microprocessor receives an external interrupt and then services that interrupt, the current program counter will be saved and an interrupt vector will be fetched. Before the program counter is saved, however, the processor will begin to fetch the next instruction of the interrupted program, but will then abort its execution. The first cycle of the aborted instruction will be labeled as an opcode by the status generator, but the second cycle will be labeled "interrupt". This helps identify that this instruction will be the first to be executed following the interrupt service routine.

80186/80188

Table J-1. 80186/80188 Interface Module Specifications (Continued)

Humidity: 90% noncondensing. Avoid sudden, extreme temperature changes which could cause condensation within the instrument.

System Configuration

Minimum hardware requirements for state analysis of an 80186/80188 target system are as follows:

- a. Model 1630A/D Logic Analyzer.
- b. 10269A General Purpose Probe Interface with Option 058 which includes:
 - (1). Model 64658A Interface Module for 80186 and 80188 including an 80186/80188 interface circuit and cable assembly.
 - (2). 80186/80188 Inverse Assembler and Format Specification setups on micro-cassette HP Part Number 10269-11006.
- c. BNC cable assembly to connect +5V accessory power from logic analyzer to general purpose probe interface.

The Model 1630D enables storage of up to 8 additional channels of information. Model 1630A cannot be used for analysis of 80186 microprocessors.

Hardware Installation

Install the interface module on the underside of the general purpose probe interface pod. Insert the free end of the interface card into the slots of the pod, then connect the pod internal cables to the interface circuit card, and fasten the cable end of the interface module to the pod using two screws.

Connect a BNC cable from the ACCESSORY POWER output on the rear panel of the logic analyzer to the ACCESSORY POWER input on the general purpose probe interface.

NOTE

Some interface modules can draw up to 1 ampere from the 5-volt supply. If using coaxial cable, HP recommends that you use RG-58, 4-foot or less to minimize voltage drop.

General Purpose
Probe Interface

CAUTION

The 80186/80188 processor may be susceptible to damage from ESD. When installing the target processor in the interface or handling the interface with the target processor installed use ESD prevention procedures.

To connect the interface cable to the target system, remove the 80186 or 80188 processor from the target system socket. Place the interface cable into the target system socket and install the U-shaped lid on the target system socket (see figure J-1). Place the processor in the interface cable socket, then install the lid on the interface cable socket.

CAUTION

The Model 64658A Interface Module is designed to mate with a Textool 268-5400 leadless chip carrier socket. The lid used to cover the interface cable when it is placed into the target system socket must be a Textool 268-5400-52 to insure proper clearance for the probe. The lid used to cover the processor in the interface cable can be any Textool compatible lid.

Connect the 1630A/D Logic Analyzer pods to the general purpose probe interface (Model 10269A) as follows:

Model 10269A Connector	Model 1630A/D Pod	
	80186	80188
A	0	1
B	1	n/c
C	2	2
D	3	3
E	4	4

NOTE

The 80188 pod to interface hookup configuration is not the standard 8 bit processor configuration. Therefore, pod to interface connections must follow the above configuration and not the configuration on the 10269A module cover.

General Purpose
Probe Interface

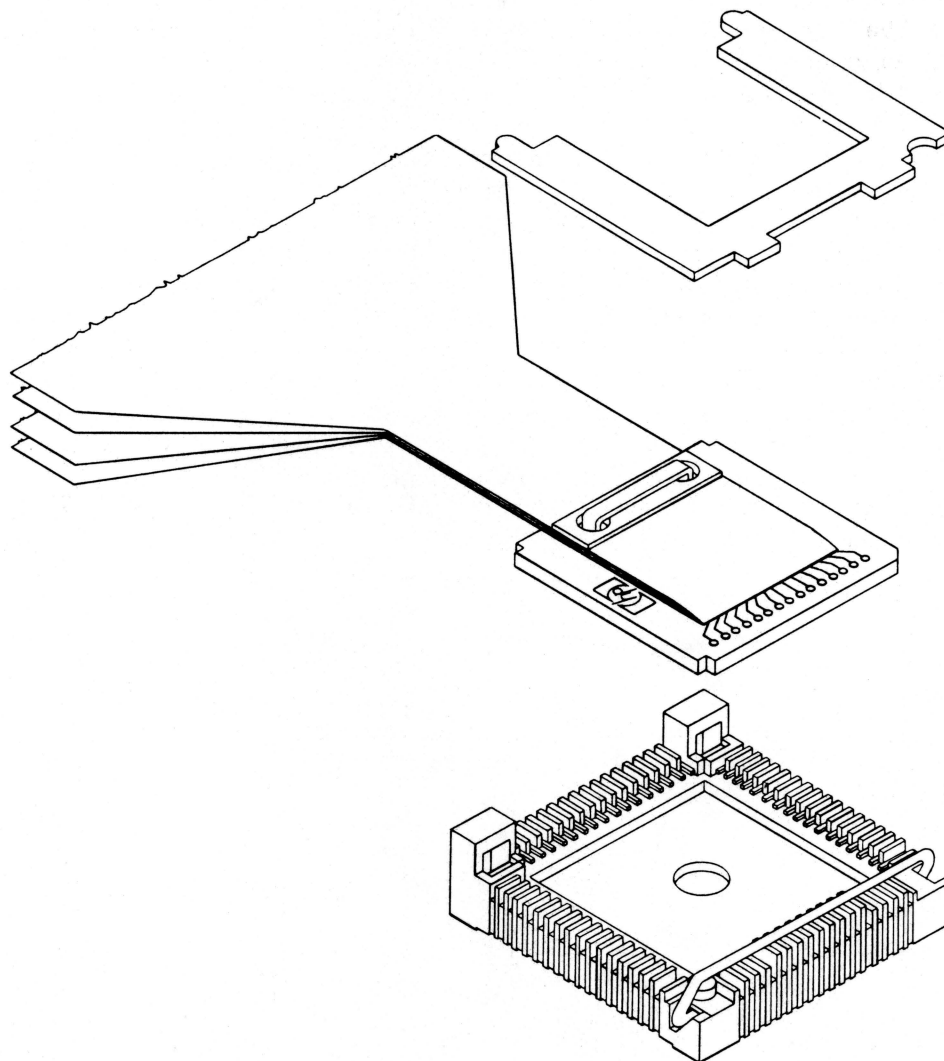


Figure J-1. Interface Cable Assembly, Exploded View

Interface Description

The primary function of an interface module is to connect the target processor through the probe interface to the logic analyzer and to perform any required functions that are unique to that particular processor. The Model 64658A Interface Module performs this primary function by (1) demultiplexing the address/data bus of the 80186/80188 processor so that both the address and data can be sent to the logic analyzer at the same time, (2) by keeping an image of the processor internal queue and program counter so that executed instructions can be captured, (3) by providing the control circuitry necessary to synchronize the executed information and the bus cycle information, and (4) by providing the ALE, RD, and WR signals to the target system when operating in the NORMAL status mode.

All address, data, and status information is collected during each processor clock (see the block diagram in figure J-2) in the bus latch. During the address portion of a bus cycle, the bus latch data is used to load the address latch or the address counter. The address latch stores the 20 bits of address and the three primary bus status lines (S0,S1,S2). In the bus mode, the address latch is loaded during all bus cycles. In the executed mode, the address latch is loaded during all non-instruction fetch cycles and the address counter is loaded during the first instruction fetch bus cycle following the status indicating that the processor queue was flushed. During the data portion of a bus cycle, the bus latch is used to provide processor data to the logic analyzer and to provide fetched instructions to the instruction queue image. In the executed mode, the data from an instruction fetch is used to load the queue image and is not supplied to the logic analyzer.

The clock to the logic analyzer is generated for two cases. First, to capture bus cycle information and, second, to capture executed instruction information. In the bus mode, only clocks for bus cycles are generated. In the execution mode, a clock is generated for each non-instruction fetch bus cycle and for each instruction byte that is executed. When the bus cycle clock is generated, the address latch and the data buffer are driving the analyzer inputs. When an executed byte clock is generated the processor queue image and the program counter are driving the analyzer.

To allow the interface module to process executed information regardless of the target system status operating mode, it is necessary to run the processor, placed in the interface cable assembly, in the QUEUE STATUS mode. This results the loss of the ALE, RD, and WR signals. The interface module cable assembly contains a circuit to provide these signals to the target system when it is operating in the NORMAL STATUS mode.

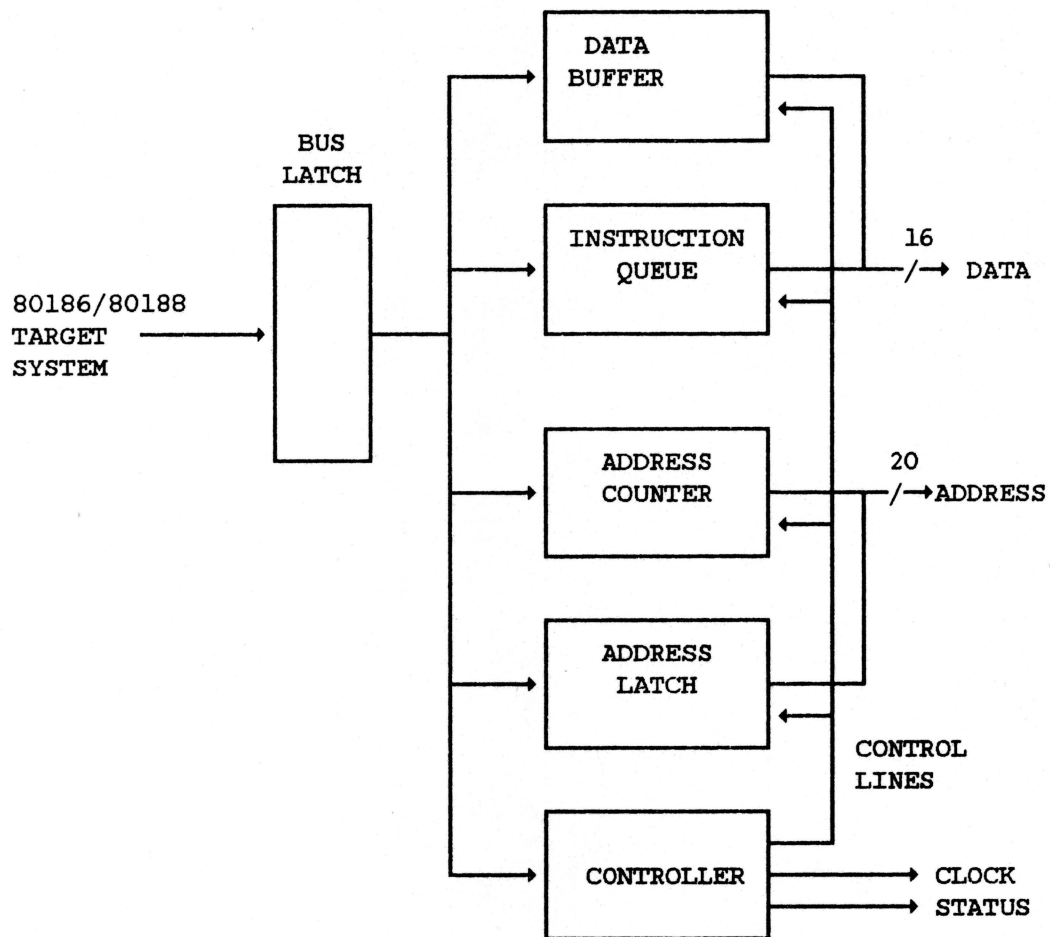


Figure J-2. 80186/80188 Interface Block Diagram

General Purpose
Probe Interface

To protect the 64658A Interface Module and the 10269A General-Purpose Interface from damage caused by static discharge, clamp diodes are included on the interface board. These diode arrays clamp the input voltage from the probe cable to the +5-volt power supply and ground of the interface. As a result, if the logic analyzer and interface are not powered up, these diode arrays will clamp the 80186/80188 signals to ground and prevent the 80186/80188 CPU from operating. If the logic analyzer cannot always be powered up when the target system is operating, the diode arrays (U2A, U5A, and U6A) can be removed from their sockets to allow operation of the CPU. It should be understood, however, that removal of the diode arrays will increase the susceptibility of the interface module to damage caused by static discharge.

Table J-2 lists the 80186/80188 signals and corresponding lines to the logic analyzer.

Table J-2. 80186/80188 Signal List

CPU SIGNAL	CPU PIN	LABEL(S)	POD	BIT
AD0 (NOTE 1)	17	ADDRESS	2	0
AD1	15	ADDRESS	2	1
AD2	13	ADDRESS	2	2
AD3	11	ADDRESS	2	3
AD4	8	ADDRESS	2	4
AD5	6	ADDRESS	2	5
AD6	4	ADDRESS	2	6
AD7	2	ADDRESS	2	7
AD8	16	ADDRESS	2	8
AD9	14	ADDRESS	3	0
AD10	12	ADDRESS	3	1
AD11	10	ADDRESS	3	2
AD12	7	ADDRESS	3	3
AD13	5	ADDRESS	3	4
AD14	3	ADDRESS	3	5
AD15	1	ADDRESS	3	6
A16/S3	68	ADDRESS	3	7
A17/S4	67	ADDRESS	3	8
A18/S5	66	ADDRESS	4	0
A19/S6	65	ADDRESS	4	1
AD0	17	DATA	0	0
AD1	15	DATA	0	1
AD2	13	DATA	0	2
AD3	11	DATA	0	3
AD4	8	DATA	0	4
AD5	6	DATA	0	5
AD6	4	DATA	0	6
AD7	2	DATA	0	7

Table J-2. 80186/80188 Signal List (Continued)

CPU SIGNAL	CPU PIN	LABEL(S)	POD	BIT
AD8 (NOTE 1)	16	DATA	1	0
AD9 (NOTE 1)	14	DATA	1	1
AD10 (NOTE 1)	12	DATA	1	2
AD11 (NOTE 1)	10	DATA	1	3
AD12 (NOTE 1)	7	DATA	1	4
AD13 (NOTE 1)	5	DATA	1	5
AD14 (NOTE 1)	3	DATA	1	6
AD15 (NOTE 1)	1	DATA	1	7
S0	52	STATUS, CPU STAT	4	2
S1	53	STATUS, CPU STAT	4	3
S2	54	STATUS, CPU STAT	4	4
BHE/S7	64	STATUS,	4	5
A19/S6	65	STATUS, SOURCE	4	6
HLDA	51	STATUS, SOURCE	4	7
(NOTE 2)		STATUS, SOURCE	4	8

Note 1: The 80186 has 16 data lines, but the 80188 only has 8 data lines; so in the case of the 80188, these lines are connected to address lines A8 through A15.

Note 2: This signal is a combination of several processor signals.

Operation

80186/80188 Selection

Because the two processors are very similar, the 64658A Interface is capable of connecting either an 80186 or an 80188 to the logic analyzer. A 16-bit versus an 8-bit data bus is the primary difference between the two processors. The processor selector switch (located at the upper-right corner of the interface board) must be set for the correct processor in order for the module to operate properly.

Normal/Queue Status Mode

Both the 80186 and the 80188 can operate in either the NORMAL or the QUEUE STATUS mode of operation. The two modes differ in what status information is available from the processor. The Model 64658A Interface Module must be configured by the user for the correct operating mode. This is done by using the switch located in the interface module cable assembly. The information presented to the logic analyzer is the same for both modes. In the bus mode, all bus cycles are sent to the logic analyzer as they occur. In the executed mode, all instruction fetch bus cycles are stored in the interface module and are sent to the logic analyzer as executed instruction bytes when they are executed. All non-instruction fetch bus cycles are sent to the logic analyzer as they occur. All coprocessor generated bus cycles on the local bus will be captured in either mode.

Coprocessor Support

The 64658A Interface Module supports coprocessors. However, the 1630D is limited in the number of pod bits available and cannot capture the CPU status signals S3, S4, and S5. Therefore, it is impossible to determine the exact source of any coprocessor memory reads or writes. For this reason, the default inverse assembler will only indicate the source of any external memory read or write with the message `"*coprocessor*"`.

Bus Cycle Mode (fetched instructions)

The Bus Cycle Mode is initiated by placing the BUS/EXECUTED switch (located on the lower-left corner of the interface board) in the BUS position. While the interface module is in this mode of operation, all bus transactions (by both the processor and coprocessors) will be sent to the logic analyzer, and the time count will be an accurate reflection of when they occurred. No distinction is made between instructions that are executed and those that are only prefetched by the processor.

Typically, there will be several states separating the memory (or I/O) transfer from the instruction that caused the transfer. The logic analyzer is clocked on the falling edge of the processor input clock at the end of the processor T₄ cycle.

Executed Instruction Mode (dequeued instructions)

The Executed Instruction Mode is initiated by placing the BUS/EXECUTED switch in the EXECUTED position. This mode provides several advantages over the bus cycle mode when tracing program flow. The most important difference is that only executed instructions are sent to the logic analyzer. Any instructions that are fetched but not executed due to program transfer conditions are not sent to the analyzer and therefore do not affect analyzer measurements. This is particularly important when performing overview measurements.

Any instruction can be used as a real-time trigger or other parameter in the trace specification, even if it starts at an odd address. For the 80186, instructions are normally fetched from even addresses unless a program transfer is made to an odd address; this makes defining the trace specification difficult if the instruction of interest is at an odd address. In the Executed Instruction Mode, however, each executed byte of code is sent to the logic analyzer with its address.

Another result of this dequeued mode of operation is that any instruction that causes either a memory or an I/O read or write will be immediately followed by that transfer. The trace list is an accurate representation of the order in which instructions are executed. The processor pulls instructions from its queue one byte at a time and executes them completely (including resultant bus cycles) before executing the next instruction. This logical ordering of instructions and data transfers further reduces possible confusion in interpreting measurement results. Executed instruction bytes are sent to the logic analyzer on the falling edge of the processor clock which is two cycles after the CPU indicates that the byte was pulled from its queue. The time count made by the analyzer is an accurate reflection of when instructions are actually executed. A bus cycle and execution of an instruction byte can occur at the same time when the target system has a coprocessor or when the 80186/80188 processor is running an internal DMA bus cycle. When this occurs the logic analyzer is supplied with the executed byte information and a flag indicating that a bus cycle has been detected.

General Purpose Probe Interface

Status Encoding

The complete status field presented to the logic analyzer by the interface module is 7 bits wide. Information in this field indicates whether the CPU is bus master for the current state or if a coprocessor is in control. It also indicates the type of bus transaction made; e.g., code fetch, memory write, interrupt acknowledge, etc.

A brief description of each of the 7 bits is given below. Bit 0 below is the least significant bit of the 7-bit field; it is brought into the analyzer on pod 4, bit 2. When reference is made to a CPU or coprocessor signal, it is of the same logic polarity as at the device pin.

Bit 0 is the device status bit LS0 for all bus cycles. For states that are CPU executed bytes, this bit = 1.

Bit 1 is the device status bit LS1 for all bus cycles. For states that are CPU executed bytes, this bit = 1.

Bit 2 is the device status bit LS2 for all bus cycles. For states that are CPU-executed bytes, this bit = 1.

Bit 3 for the 80186 and coprocessors is LBHE; it is low when a transfer is made on the high byte of the data bus. For the 80188, it is always high. For executed bytes, it is low for the execution of a first byte and high for the execution of a subsequent byte. For halt acknowledge cycles, this bit is undefined.

Bit 4 is the device status bit S6 and indicates whether the current cycle is a DMA or non-DMA cycle, or whether it is a coprocessor cycle. For states that are CPU-executed bytes, this bit = 0. For halt acknowledge cycles, this bit is undefined.

Bit 5 is the device status HLDA for all bus cycles. For states that are CPU-executed bytes, this bit = 0.

Bit 6 is used to indicate that the current state is a bus cycle. If this bit occurs and the LS0, LS1, LS2 status indicates a CPU-executed byte, this bit flags a lost bus cycle.

The 7-bit status field is labeled STATUS in the format specification and is used by the inverse assembler. This field is not associated with a symbol map, so the precise value of each bit is shown in table J-3.

Table J-3. STATUS Field Encoding

80186 or 80188 CYCLES	STATUS bit						
	6	5	4	3	2	1	0
Executed byte	x	0	0	x	1	1	1
Executed first byte of instruction	x	0	0	0	1	1	1
Executed next byte of instruction	x	0	0	1	1	1	1
Executed byte - lost bus cycle	1	0	0	x	1	1	1
Bus cycle	1	0	x	x	x	x	x
Interrupt Acknowledge	1	0	0	x	0	0	0
Write I/O port - non dma	1	0	0	x	0	1	0
Read I/O port - non dma	1	0	0	x	0	0	1
Halt Acknowledge	1	0	x	x	0	1	1
Code fetch	1	0	0	x	1	0	0
Write memory - non dma	1	0	0	x	1	1	0
Read memory - non dma	1	0	0	x	1	0	1
Read I/O port - dma	1	0	1	x	0	0	1
Write I/O port - dma	1	0	1	x	0	1	0
Read memory - dma	1	0	1	x	1	0	1
Write memory - dma	1	0	1	x	1	1	0
COPROCESSOR CYCLES	STATUS bit						
	6	5	4	3	2	1	0
Word transfer (A0 = 0)	1	x	x	0	x	x	x
High byte transfer (A0 = 1)	1	x	x	0	x	x	x
Low byte transfer (A0 = 0)	1	x	x	1	x	x	x

Format Specifications

The 80186 Inverse Assembler file and the 80188 Inverse Assembler file contain predefined format specifications (see figures J-3 and J-4). These format specifications include all labels that are needed to monitor the 80186/80188 CPU and any coprocessors that are connected directly to the CPU.

State Format Specification _____		Inverse Assembler Loaded _____			
[Assignment]		Clock			
		JKL			
		[.↓.]			
Multiplexing	Pod4	Pod3	Pod2	Pod1	Pod0
[Off]	[TTL]	[TTL]	[TTL]	[TTL]	[TTL]
	8.....0	8.....0	8.....0	7.....0	7.....0
Activity>					
Label	Pol				
ADDR	[+]	[.....** *****]			
DATA	[+]	[.....]	[*****]		
STAT	[+]	[*****]			

Figure J-3. 80186 Format Specification

State Format Specification _____		Inverse Assembler Loaded _____			
[Assignment]		Clock			
		JKL			
		[.↓.]			
Multiplexing	Pod4	Pod3	Pod2	Pod1	Pod0
[Off]	[TTL]	[TTL]	[TTL]	[TTL]	[TTL]
	8.....0	8.....0	8.....0	7.....0	7.....0
Activity>					
Label	Pol				
ADDR	[+]	[.....** *****]			
DATA	[+]	[.....]	[*****]		
STAT	[+]	[*****]			

Figure J-4. 80188 Format Specification

The 80186/80188 Inverse Assembler

The inverse assembler is designed to support the 80186 and 80188 processors with the BUS/EXECUTED switch on the interface module in either position. The following paragraphs explain inverse assembler operation and conditions where certain results can be expected.

Unless followed by a lower-case letter, all numeric output from the inverse assembler can be assumed to be in hexadecimal format. A lower-case "o" following a numeric value indicates an octal representation (the ESC instruction for example). Decimal values are indicated by a lower-case "d" (as in the INT instruction). When monitoring an 80186 processor when the BUS/EXECUTED switch on the interface module is set to the BUS position, two instructions may be displayed for a single analyzer state. This is possible because the 80186 fetches a word containing two instruction bytes from program memory. If the least significant byte of this word contains a single byte instruction, the next sequential instruction begins in the upper byte. In this case, the two instructions displayed on a single line are separated by the delete symbol (█). Since instructions may begin in either the lower or upper byte, it is also possible for the last byte of a multiple byte instruction to occur in the lower byte, with a second instruction beginning in the upper byte. In this instance, the delete symbol is displayed in the left-most position of the mnemonic display field. Thus, the following definition: Any instruction appearing to the right of the delete symbol begins in the upper byte of the fetched word. Three examples are shown as follows:

PUSH DX █ ADC BX,DX	(PUSH occupies the lower byte; ADC begins in the upper byte.)
█ CMP AX,#53E6	(An instruction shown on a previous line uses the lower byte; CMP begins in the upper byte.)
JO OFLOW_CTL	(JO begins in the lower byte, and uses the upper byte as well.)

The appearance of asterisks in the inverse assembler output is an indication that a portion (or portions) of an instruction was not captured by the analyzer. Missing opcodes occur frequently when the BUS/EXECUTED switch is placed in the BUS position; this is primarily due to processor prefetch activity. Storage qualification, or the use of storage windows, can also lead to such occurrences.

General Purpose
Probe Interface

The 80186 processor can perform word transfers as well as byte transfers between processor registers and memory. Furthermore, byte transfers may occur either on the upper 8 bits, or the lower 8 bits of the 16-bit data bus. The inverse assembler makes a distinction between these conditions by displaying "xx" (don't care) for the byte of the transfer that was ignored by the processor. In this way, it is possible to determine exactly which byte was utilized by the processor:

28B3 memory write	(word transfer)
xxB3 memory write	(byte transfer on lower 8 bits)
28xx memory write	(byte transfer on upper 8 bits)

The 80186/80188 instruction set contains four groups of instructions where the actual instruction type is defined in the second opcode rather than in the first. In this case, if the second opcode is not stored within analyzer memory, it is possible to determine only the group in which an instruction resides. Therefore, the group name is displayed in the mnemonic display field rather than an instruction mnemonic. These group names are defined as follows:

Immed - Contains the following instructions when used with immediate source operands:

ADD	AND
OR	SUB
ADC	XOR
SBB	CMP

Shift - Contains the following logical and arithmetic shifts and rotates:

ROL	SHL/SAL
ROR	SHR
RCL	SAR
RCR	

Grp_1 - Contains the following instructions. The TEST instruction is included only when dealing with an immediate source operand:

TEST	IMUL
NOT	DIV
NEG	IDIV
MUL	

Grp_2 - Contains the following three groups of instructions.

INC	}	when dealing with memory operands on 8-bit registers
DEC		
CALL	}	indirect operand
JMP		

PUSH when dealing with 16-bit memory operands

Listed below are several abbreviations for normal programming syntax that have been adopted in order to reduce the width of the inverse assembler display field. For further information and a description of these keywords, refer to the 64000 Logic Development System 80186 Assembler Supplement.

dwp	...	DWORD	PTR
wp		WORD	PTR
bp		BYTE	PTR
fp		FAR	PTR
np		NEAR	PTR
s		SHORT	

It should be noted that these symbols are displayed only if it is not possible to determine the operation size from the instruction itself.

To further reduce inverse assembler field width, LOCK and REPEAT prefixes appear on the line before the instruction to which they apply.

Physical, rather than logical addresses are used to perform symbolic address mapping. Most instructions, however, specify a 16-bit intrasegment offset and may optionally indicate a segment different from the default segment for that particular instruction. Since it is not possible to determine the physical address from this information alone, the inverse assembler must attempt to locate the actual resultant bus cycle so that the physical address may be obtained. If a bus cycle of the type indicated by the initiating instruction is not found, the physical address cannot be determined and an unmapped logical address (segment override, if any, and the 16-bit intrasegment offset) is displayed instead of a mapped physical address.

General Purpose
Probe Interface

Because resultant bus cycles must therefore be located within the trace list, successful address mapping depends upon judicious use of analyzer storage qualification capabilities. Since it is not possible to locate bus cycles corresponding to specific instructions when operating with the BUS/EXECUTED switch in the BUS position, address mapping is possible only with the the BUS/EXECUTED switch in the EXECUTED position.

When operating with a coprocessor such as the 8087 or 8089, it is possible that the processor can extract a byte from its queue at the same time that the coprocessor is running a bus cycle. In this case, the executed byte, rather than the coprocessor cycle, is sent to the analyzer. The inverse assembler indicates this lost bus cycle condition by displaying "lbc" in the right-most positions of the mnemonic display field.

